

C L A I M S

1. A method of executing software code of a software program on an external unit, the software code being parsed into a plurality of different blocks of code, each block of code being independently executable, wherein the external unit is in communication with a computer, the computer including memory for holding the software code, the external unit including (i) input/output for communication with the computer, (ii) a processor, and (iii) memory, the method comprising:
- 5 (a) automatically uploading a first block of code to the memory of the external unit;
- 10 (b) executing the first block of code in the external unit using only the processor and the memory of the external unit; and
- (c) sequentially and automatically uploading and executing the remaining blocks of code of the software code in the external unit, wherein subsequent blocks of code overwrite previously uploaded blocks of code in the memory of the external unit.
- 15
2. The method of claim 1 wherein the software code is a smart card application (applet), and each of the different blocks of code are functions or methods.
3. The method of claim 2 wherein the computer includes a plurality of software programs and applets, and steps (a) and (b) are performed for each applet.
- 20
4. The method of claim 1 wherein step (a) is performed in the computer at run-time of the software code.
5. The method of claim 1 wherein step (a) is performed prior to runtime of the software code.
- 25
6. The method of claim 1 wherein the software code includes a first portion and a second portion, the second portion being the software code having a plurality of different blocks of code, each block of code being independently executable, the method further comprising:
- 30 (d) executing the first portion of code in the computer.

7. The method of claim 1 wherein the software code is encrypted, the method further comprising:

(d) decrypting each block of code in the external unit prior to execution.

5 8. The method of claim 1 further comprising:

(d) after execution of the last block of code, the external unit sending back state information to the computer for subsequent use by at least the first portion of the software code.

10 9. The method of claim 1 wherein the external unit is a smart card.

10. A method of executing software code of at least one software program in a multi-processor computer environment, each software program including (i) a first portion of software code to be executed in a computer, and (ii) a second portion of software code to be executed in one or more external units which are in communication with the computer, the software code of the second portion being parsed into a plurality of different independently executable blocks of code, each external unit including (i) input/output for communication with the computer, (ii) a processor, and (iii) memory, the method comprising:

20 (a) automatically uploading a first block of code to the memory of an external unit at execution time of the second portion of software code;

(b) executing the first block of code in the external unit using only the processor and the memory of the external unit; and

25 (c) sequentially and automatically uploading and executing the remaining blocks of code of the software code in the external unit, wherein subsequent blocks of code overwrite previously uploaded blocks of code in the memory of the external unit.

30 11. The method of claim 10 wherein the second portion of software code is a smart card application (applet), and each of the different blocks of code are functions or methods.

12. The method of claim 11 wherein the software program includes a plurality of applets interspersed within the software program, and steps (a)-(c) are performed for each applet.

5 13. The method of claim 10 wherein the second portion of software code is encrypted, the method further comprising:

(d) decrypting each block of code in the external unit prior to execution.

14. The method of claim 10 further comprising:

10 (d) after execution of the software code, the external unit sending back state information to the computer for subsequent use by at least the first portion of software code.

15. The method of claim 10 wherein the external unit is a smart card.

15

16. A method of preparing software code of a software program to be executed on an external unit which is in communication with a computer, the computer including memory for storing the software code, the method comprising parsing the software code into a plurality of different blocks of code which can be sequentially
20 uploaded to, and independently executed in, the external unit.

17. The method of claim 16 wherein the software code is a smart card application (applet), and each of the different blocks of code are functions or methods.

25 18. The method of claim 16 wherein the software program includes (i) a first portion of software code to be executed in the computer, and (ii) a second portion of software code to be executed in the external unit which is in communication with the computer, and only the second portion of software code is parsed into a plurality of different blocks of code which can be sequentially uploaded to, and inde-
30 pendently executed in, the external unit.

19. A method of preparing a source code program comprising creating pre-compiled source code from original source code, wherein at least a portion of the pre-compiled source code is source code having a function call with arguments

that are encrypted machine code, and the pre-compiled source code has the same language syntax as the original source code.

20. The method of claim 19 wherein the pre-compiled source code includes interspersed first and second portions of pre-compiled source code, only the second portions being source code having a function call with arguments that are encrypted machine code.

21. A method of preparing protected computer code from original source code of a software program, the original source code including interspersed first portions and second portions of code of the software program, the method comprising creating a pre-compiled version of the original source code by:

(a) transforming each second portion into a function call with arguments that are encrypted executable machine code; and

(b) copying each first portion to the pre-compiled version,
the pre-compiled version of the original source code having the same language syntax as the original source code.

22. The method of claim 21 wherein each of the second portions of the original source code original block of code has an associated tag, and step (a) uses the tags to identify each second portion for transformation.

23. The method of claim 21 further comprising:

(c) re-compiling the precompiled version of the original source code into a single integrated executable machine code program having function calls which are associated with the encrypted executable machine code.

24. A method of executing one or more blocks of protected software code within a machine code program in a plural processor environment, each block of protected software code having a function call with arguments that are encrypted executable machine code, the method comprising:

(a) executing at least portions of the machine code program in a first processor; and

(b) upon reaching a function call for a block of protected software code, decrypting and executing the associated protected software code in a second processor.

5 25. The method of claim 24 wherein step (b) further comprises upon reaching a function call for a protected block of software code, sending the associated protected software code to the second processor for decryption and execution therein.

26. The method of claim 24 further comprising:

10 (c) upon initiation of step (a), uploading all of the blocks of protected software code to a memory associated with the second processor for subsequent decryption and execution therein upon reaching each of the respective function calls.

27. The method of claim 24 wherein the second processor is a smart card.